GNNs for tracking
(specifically: for track pattern recognition)

Jan Stark
Laboratoire des 2 Infinis – Toulouse (L2IT)

ACTS ML meeting, February 26th 2021

What we are working on

Goal: develop a realistic GNN-based algorithm for pattern recognition that can be deployed in an experiment at the HL-LHC (i.e. that is able to deal with complex detector geometry, lots of pile-up, large events, …).
Implement this algorithm in ACTS.
Run it concurrently with other algorithms (e.g. Kalman filter) on exactly the same events.

Towards a realistic track reconstruction algorithm based on graph neural networks for the HL-LHC

Catherine Biscarat[1], *Sylvain* Caillou[1], *Charline* Rougier[1], *Jan* Stark[1], and *Jad* Zahreddine[1]

[1]Laboratoire des 2 Infinis - Toulouse (L2IT-IN2P3), Université de Toulouse, CNRS, UPS, F-31062 Toulouse Cedex 9, France

Abstract. The physics reach of the HL-LHC will be limited by how efficiently the experiments can use the available computing resources, i.e. affordable software and computing are essential. The development of novel methods for charged particle reconstruction at the HL-LHC incorporating machine learning techniques or based entirely on machine learning is a vibrant area of research. In the past two years, algorithms for track pattern recognition based on graph neural networks (GNNs) have emerged as a particularly promising approach. Previous work mainly aimed at establishing proof of principle. In the present document we describe new algorithms, implemented in the ACTS framework, that can handle complex realistic detectors. This work aims at implementing a realistic GNN-based algorithm that can be deployed in an HL-LHC experiment.

Paper about to be submitted to vCHEP (this Sunday). Will also submit to arXiv.

Considerations for integration into the ACTS repository

Graphs represented using classes from boost. ACTS already depends on boost.

Graphs can be written to disk for GNN training (outside ACTS, e.g. in TensorFlow).

… or can be kept in memory for track reconstruction (“inference”).

ACTS classes for graph creation, output to disk have been implemented (and used for the vCHEP paper).
They could simply be added to the ACTS repository “in a new subdirectory”).
Do not introduce any new dependency to ACTS.

TensorFlow code for GNN training has been implemented (and used for the vCHEP paper).
Would be nice to also store it in the ACTS repository, even if it is completely independent of the ACTS framework.

Code for inference has been written outside ACTS (and used for the vCHEP paper). Inference not ported to ACTS yet.
Could create a new “example” in the ACTS repository that is not compiled/linked by default. Might introduce new dependencies to ACTS.